

SFX e OpenURL: gli esperimenti del team di Van De Sompel e l'evoluzione dello standard

di Cinzia Bucchioni

(Pisa - Bologna 2002)

Roma 10 aprile 2003

Sommario

- La sperimentazione SFX
 - le basi teoriche
 - i nodi tecnici
 - la terminologia
- Lo standard OpenURL
 - l'integrazione col DOI
 - la sua evoluzione
- L'openResolver
- Conclusioni

Riferimenti essenziali

- **I cinque articoli di Van De Sompel et al. sul “D-Lib Magazine” 1999-2001**
<http://mirrored.ukoln.ac.uk/lis-journals/dlib/dlib/dlib.html>
e le traduzioni di essi sul sito della CNUR:
<http://www.aib.it/aib/commiss/cnur/scaffale.htm3>
- **La documentazione sulla OpenURL**
 - Sintassi v.0.1 <http://www.sfxit.com/openurl/openurl.html>
 - Sito dedicato alla OpenURL: <http://library.caltech.edu/openurl/>
 - Nuovo draft dello standard
 - <http://library.caltech.edu/openurl/PubComDocs/StdDocs/Part1-PC-20030401.pdf>
 - <http://library.caltech.edu/openurl/PubComDocs/StdDocs/Part2-PC-20030401.pdf>
- **Articoli vari**
 - Relazione di Miriam Blake sul beta test a Los Alamos
(http://lib-www.lanl.gov/lww/articles/OpenURL_vala.pdf)
e relative slides per il Convegno di Vala
(<http://www.sfxit.com/publications/presentations/vala2002v2.ppt>)
 - slides di D. Flecker per il Convegno Nelineet
(<http://www.nelineet.net/calendar/seminars/flecker.ppt>)
 - *Open Citation Linking* di S. Hitchcock et al. sul “D-Lib Magazine” ottobre 2002
 - *OpenResolver: a simple OpenURL resolver* di Andy Powell su “Ariadne” giugno 2001
<http://www.ariadne.ac.uk/issue28/resolver/>
 - *The UPS Prototype* di Van De Sompel et al. sul “D-Lib Magazine” del febbraio 2000
- **Tutta la documentazione del sito SFX (Ex Libris)**
<http://www.sfxit.com/publications/>

La sperimentazione ha prodotto:

- un prototipo software di componente di servizio
 - ◆ acquistato, ingegnerizzato e commercializzato da Ex Libris: SFX server
- la proposta dello standard OpenUrl alla NISO
- un bagaglio di esperienze, conoscenze, riflessioni teoriche, che hanno permesso elaborazioni analoghe:
 - ◆ il modulo *free sw* dell'OpenResolver di UKOLN e il Godot Free sw
 - ◆ prodotti di diversi fornitori commerciali
LinkFinderPlus (Endeavor, OCLC e Ebsco),
Fretwell-Downing (integrato in Zportal), *Knowledge Info-Net Ser*,
Vlink sviluppato a Brussel della GEAC, etc.

Il nome SFX

- È nome utilizzato durante la sperimentazione, che ha condotto al software prototipale
- Ora definisce uno specifico prodotto commerciale
- Altri prodotti software analoghi vengono commercializzati con altri nomi
- Nomi più generici per questo tipo di tecnologia:
 - Risolutore di openURL
 - Componente Istituzionale di Servizio (ISC)

La sperimentazione

Una tecnologia per interconnettere le risorse elettroniche in una biblioteca ibrida

- Creare un ambiente digitale accademico
 - completamente interconnesso
 - collegando tutte le risorse
 - Porre la biblioteca digitale/ibrida in grado di strutturare tali collegamenti
 - secondo le esigenze locali
 - secondo la collezione locale
- => servizio di linking localizzato o *context sensitive*

L'iter della sperimentazione

- Definizione di servizi estesi
 - strutturali di un ambiente informativo pienamente interconnesso
- Riflessione sui sistemi di linking
 - collegamenti localizzati (sotto il controllo della biblioteca)
- realizzazione di un'architettura sw appropriata
 - generalizzazione di essa

I servizi estesi

- Cerca gli autori in altri databases
- controlla quanto sono citati gli autori
- trova l'indirizzo email degli autori

Possiede la mia biblioteca
copia cartacea della rivista?

Bookmarks Location: <http://www.iop.org/EJ/S/UNREG/ATE2dYQA4jUXXBUPgvG6jA/article/0305-4470/33/35/101/A035I> What's Relat

References

- [1] Chodos A 1980 *Phys. Rev. D* **21** 2818
Olive D and Turok N 1985 *Nucl. Phys. B* **257** 277
Eguchi T and Yang S K 1989 *Phys. Lett. B* **224** 373
- [2] Drinfel'd V G and Sokolov V V 1985 *J. Sov. Math.* **30** 1975
Drinfel'd V G and Sokolov V V 1981 *Sov. Math. Dokl.* **23** 457
- [3] Aratyn H, Ferreira L A, Gomes J F and Zimerman A H 1997 *J. Math. Phys.* **38** 1559
(Aratyn H, Ferreira L A, Gomes J F and Zimerman A H 1995 *Preprint hep-th/9509096*)
- [4] Gomes J F, Gueuvoghlianian E P, da Silveira F E M, Sotkov G M and Zimerman A H 2000 *Singular conformal and conformal affine non-Abelian Toda theories M V Saveliev Memorial V*
Gomes J F, Gueuvoghlianian E P, Sotkov G M and Zimerman A H 1999 *press*
(Gomes J F, Gueuvoghlianian E P, Sotkov G M and Zimerman A H 2000 *Preprint hep-th/9902173*)
Gomes J F, Sotkov G M and Zimerman A H 1998 *Phys. Lett. B* **435** 49
(Gomes J F, Sotkov G M and Zimerman A H 1998 *Preprint hep-th/9803122*)
Gomes J F, Sotkov G M and Zimerman A H 1999 *Ann. Phys., NY* **274** 289-362

Full Text
appropriato

Se il full-text non è disponibile,
c'è servizio di document delivery?

Terminologia relativa ai servizi estesi

Origin

la risorsa da cui parte una ricerca e in cui si trova una citazione

link-source

il record (o comunque la citazione) che rappresenta un'entità bibliografica, per cui si vogliono servizi estesi di linking

target

la risorsa/il record a cui conduce il link che parte dalla link-source

componente di servizio

il modulo software che fornisce servizi estesi, sovrapposti ai dati dall'esterno

Per farsi un'idea concreta
di servizi estesi effettivi

<http://library.caltech.edu/openurl/Demos.htm>

<http://www.ukoln.ac.uk/distributed-systems/openurl/>

Servizi estesi = collegamenti => sistema di linking

- invece di un linking diretto ed univoco



- un linking indiretto, multiplo, **localizzato**



Linking sotto la responsabilità del fornitore di servizi

- per sottrarre i link ai possessori dei contenuti
 - sono di ambito limitato
- per valorizzare ed integrare le risorse locali

Breve storia dei link /1

- **link statici (i più semplici)**



!!! vanno costruiti e aggiornati a mano
per ogni singolo oggetto !!!

Breve storia dei link /2

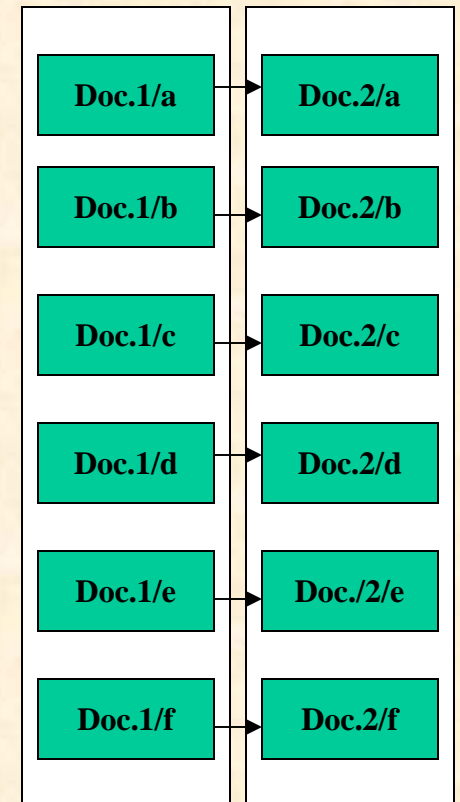
- **link algoritmici**

calcolati sulla base delle “regole” con cui ogni target costruisce la URL di accesso ai propri elementi informativi

=> i primi servizi di linking tra DB bibliografici e testi pieni

!!! non tutti i target hanno url calcolabili
cioè regolari

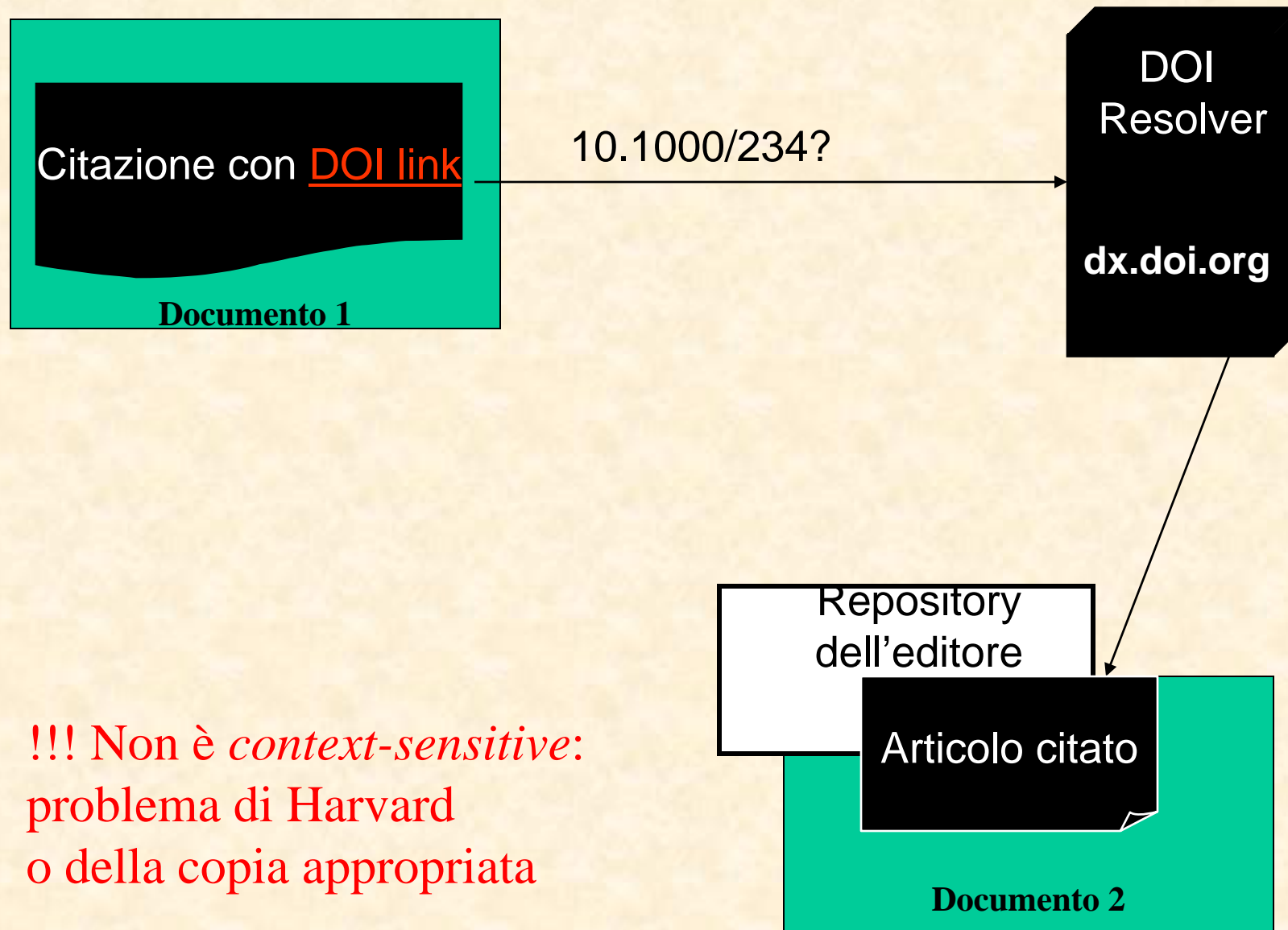
!!! Si tratta pur sempre di soluzioni ad hoc
da una certa risorsa d'origine ad una certa
risorsa target



Breve storia dei link /3

- **Assegnare un identificatore unico alle risorse**
--> servizio di risoluzione centrale
 - p.e. il DOI, nato da una collaborazione tra editori per collegare citazioni e full-text di articoli
 - funziona grazie a server centrali, dove viene registrato, per ogni nuovo articolo:
 - i metadati (la descrizione)
 - il codice DOI
 - la url del full-text (sul server dell'editore)

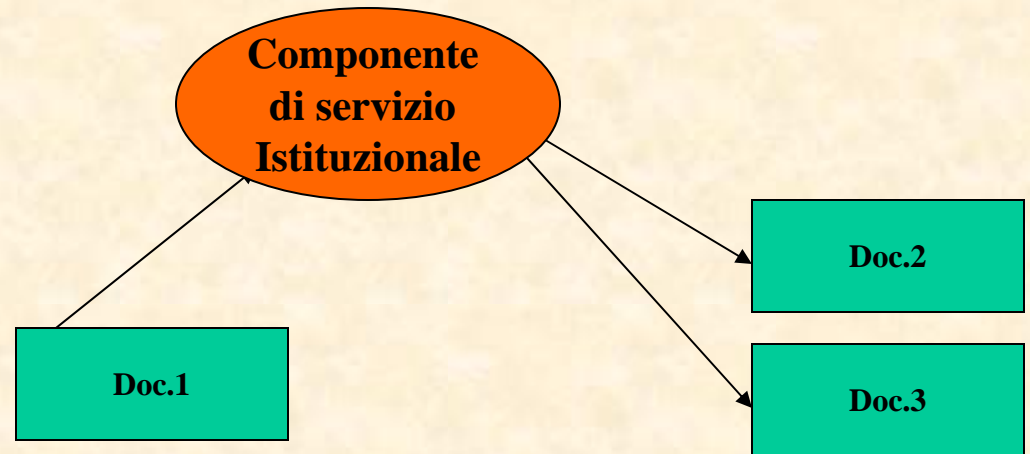
Il sistema DOI/CrossRef



Un passo ulteriore nel concetto di linking

- link algoritmici ma dinamici
 - non precalcolati in un db di link tra documenti, ma basati su db di interrelazioni potenziali
- Link aperti
 - context sensitive, non predeterminati
- link just in time
 - calcolati al momento del click e non prima, per alleggerire la rete

Un Software per realizzare questi collegamenti deve essere in grado di:



- ottenere i metadati dalla *link-source* (OpenURL)
- elaborarli per individuare e selezionare i *target* opportuni
 - ha bisogno di una “rappresentazione” della situazione contestuale per poter individuare e selezionare le relazioni potenzialmente rilevanti tra risorse (=> la *Knowledge Base*)
- saper accedere ai *target* selezionati

Knowledge Base

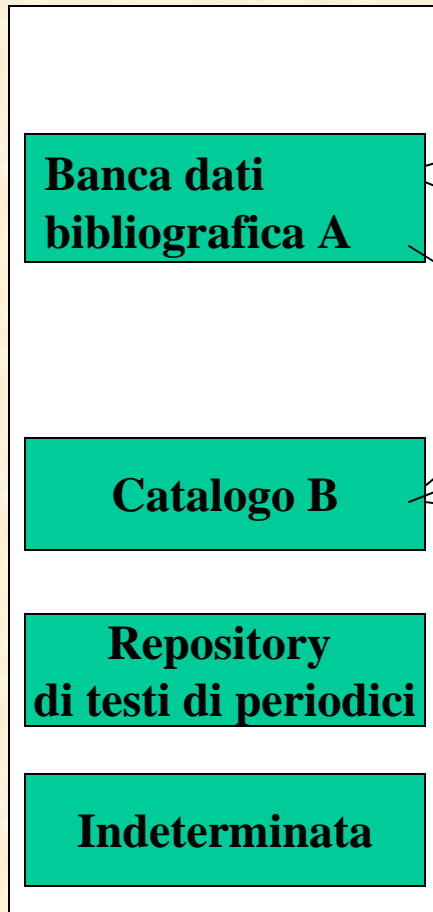
- **Repertorio di conoscenza contestuale**
(risorse, singoli oggetti in esse, relazioni, regole etc.)
 - stabilisce **relazione potenziali tra risorse**
 - **contiene le soglie di attivazione** per ogni link, in base ai valori dei parametri della link-source
 - **globali**
 - **locali**

Architettura della BASE SFX

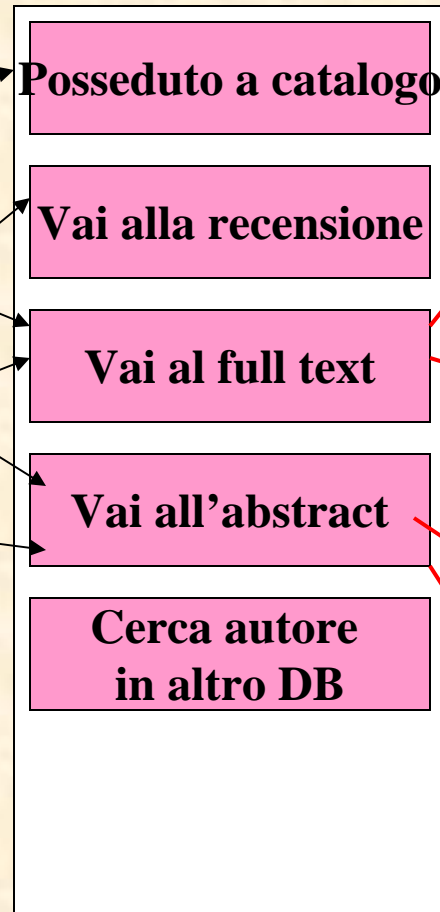


Cercando di capire:

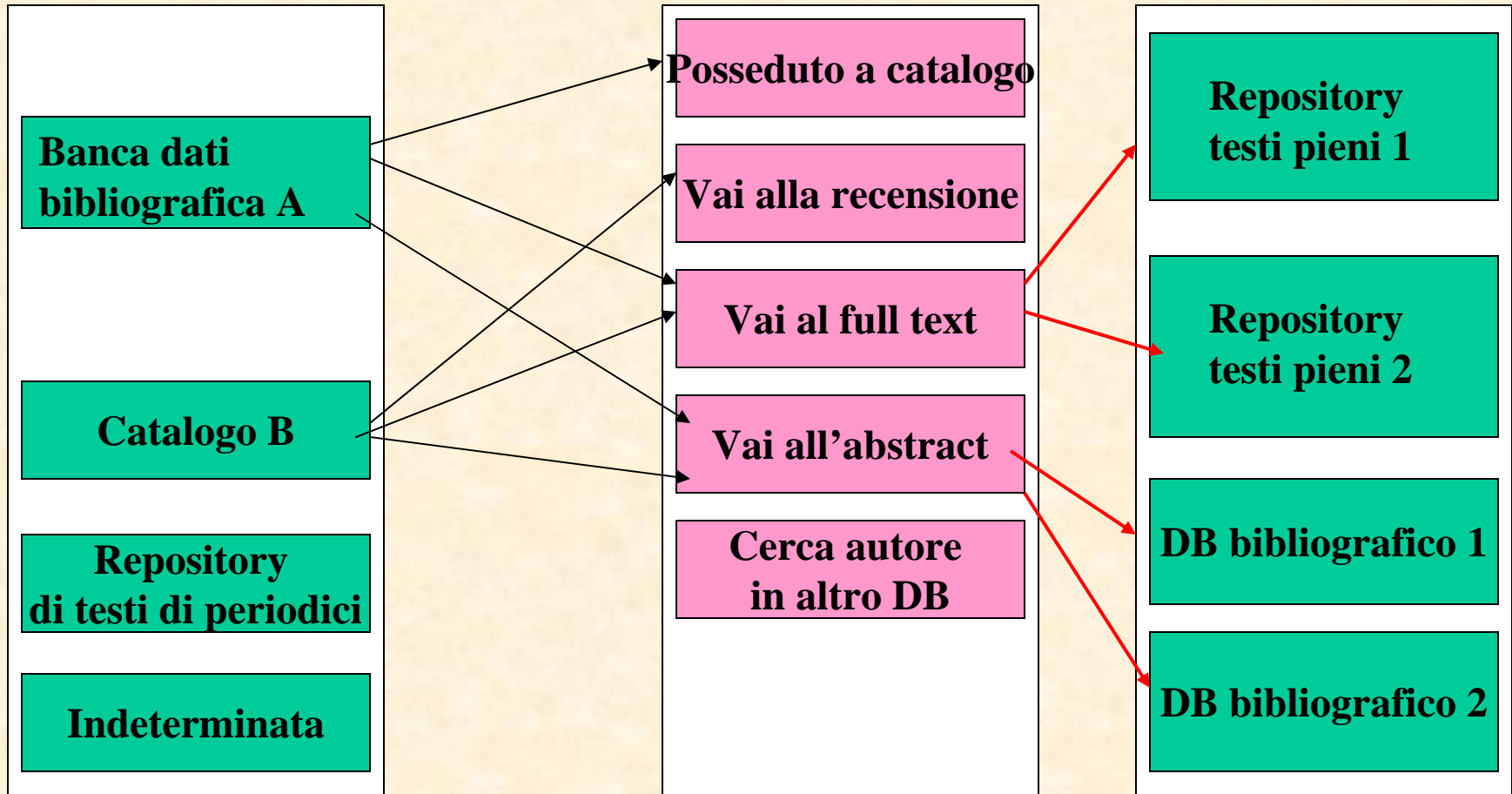
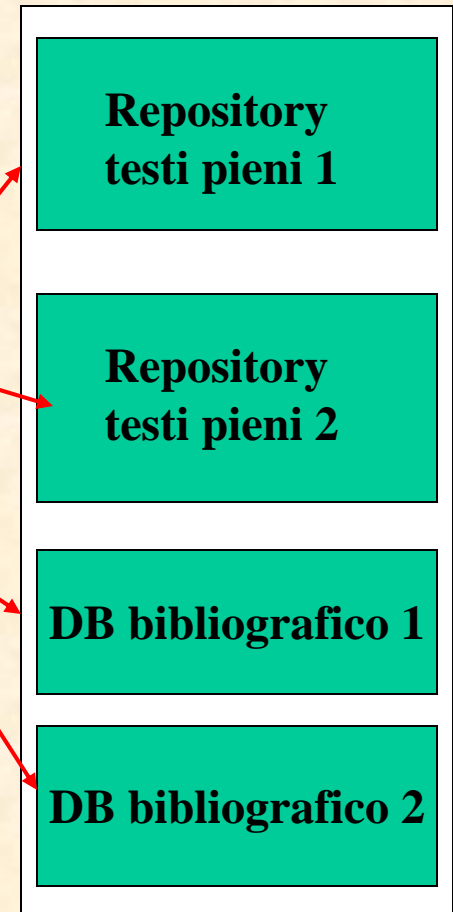
Origin/Sources



Servizi



Target

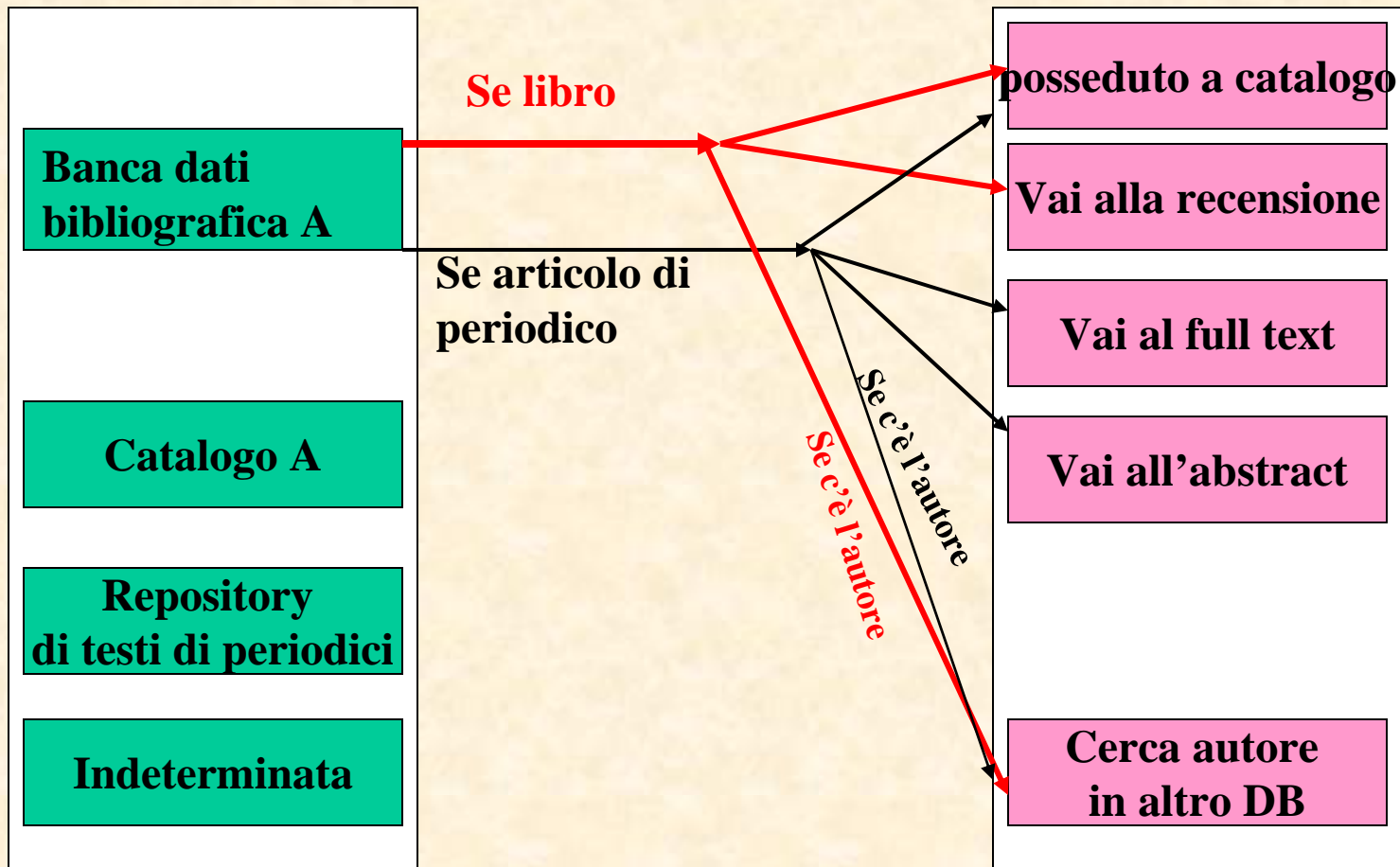


Cercando di capire /2:

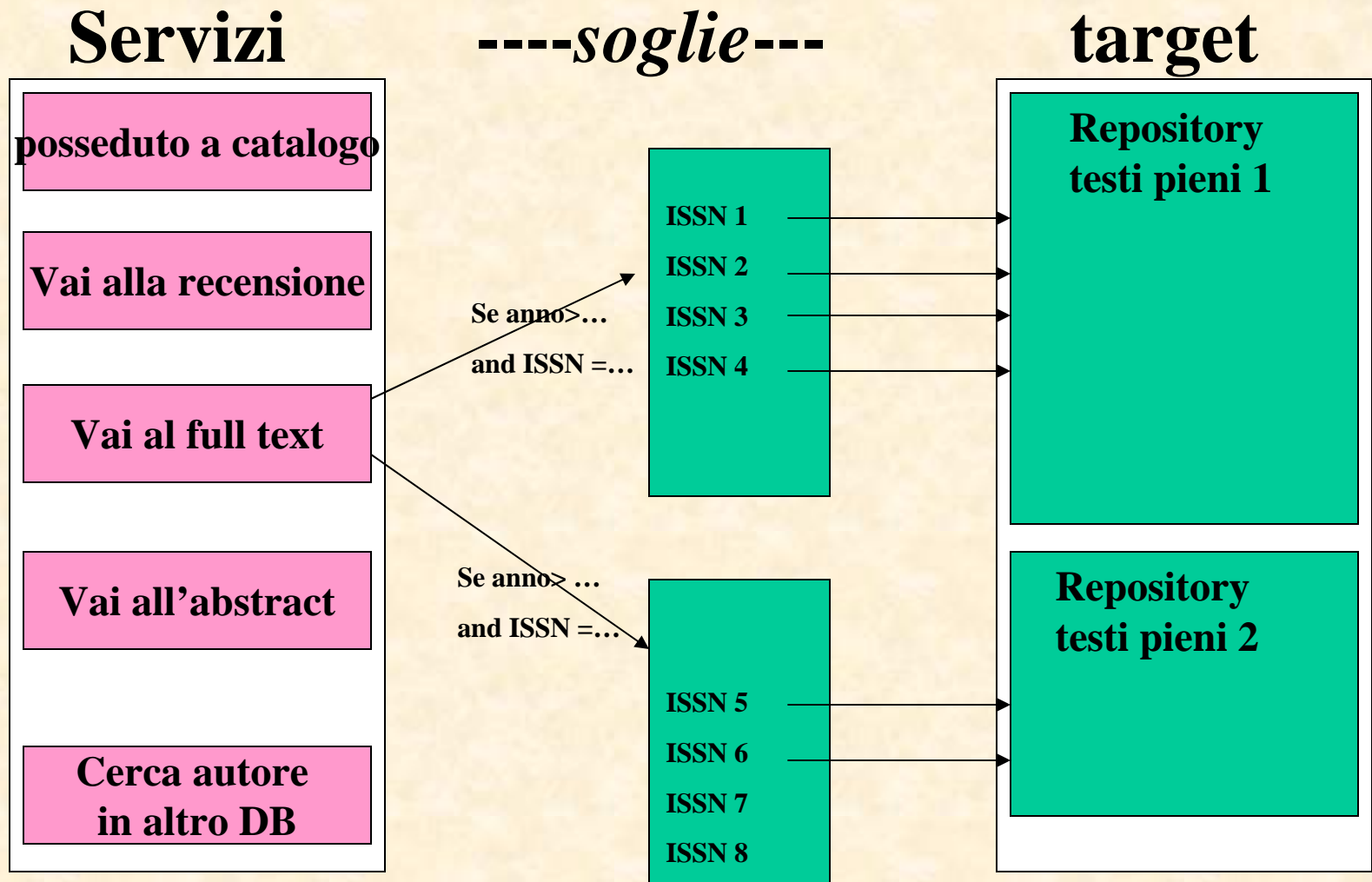
Origin/Sources

soglie

Servizi



Cercando di capire /3:



Come raggiungere i target

- soluzione pragmatica basata su script targetParser specifici per risorsa
 - sono piuttosto semplici da costruire
 - implementano la sintassi link-to
 - oppure lanciano una ricerca

Generalizzare un'applicazione

- modularizzarla
 - moduli sw indipendenti e portabili
 - che colloquino per precisi input e output
 - per cui sia facile aggiungere risorse
 - per cui sia facile cambiare ambiente

=> sperimentazione in due fasi, ampliando l'ambito di applicazione:

1. Elektron: solo le risorse dell'Università di Ghent
2. Ghent&LANL: Università di Ghent e di Los Alamos

Un SW modulare

targetParser

uno script (un pezzo di sw) che rientra nell'architettura del componente di servizio, capace di implementare il collegamento all'interno di un particolare target (una part. sintassi link-to)

sourceParser

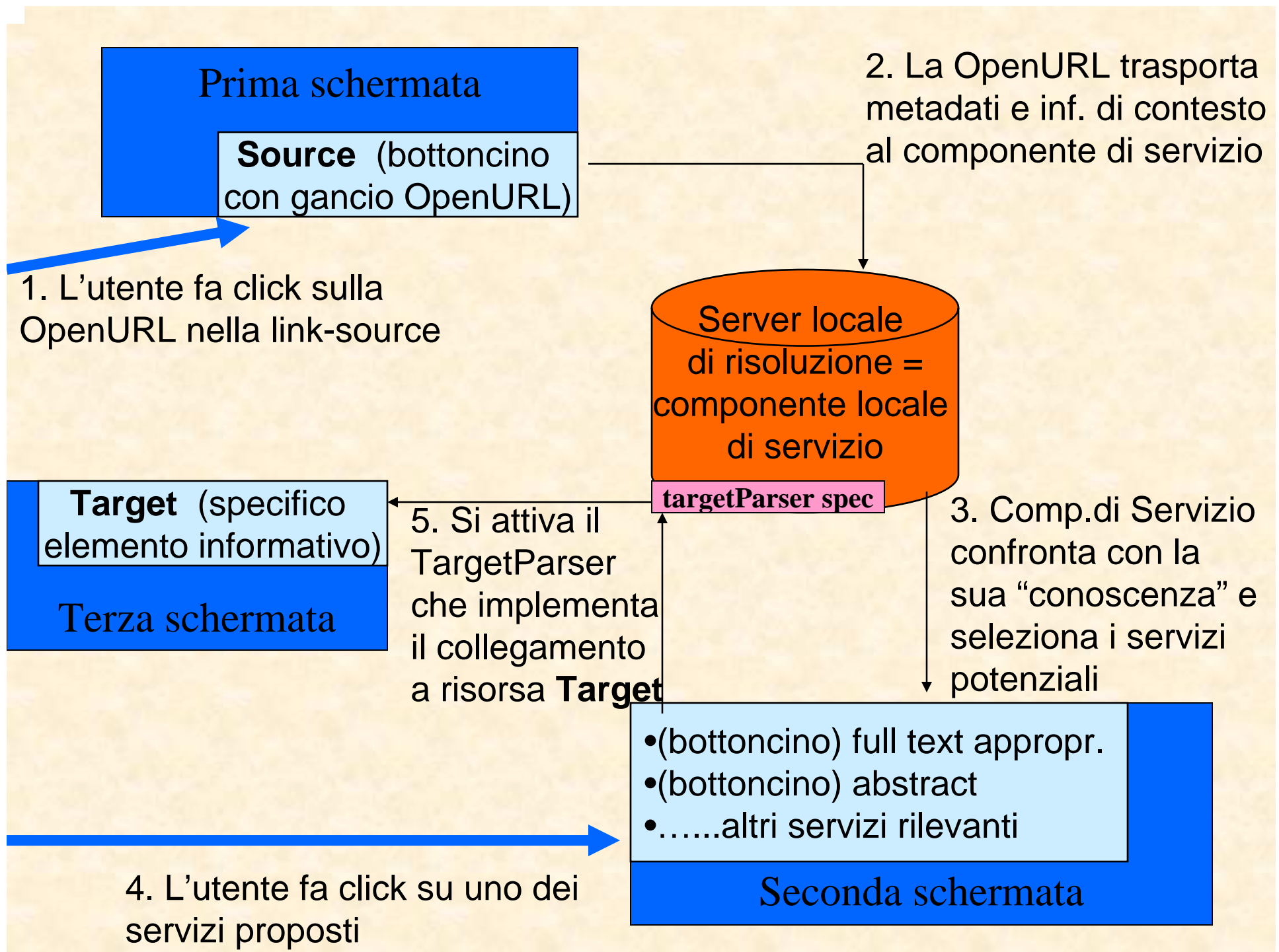
uno script (un pezzo di sw) che rientra nell'architettura del componente di servizio, capace di analizzare una particolare link-source)

Come ricevere i metadati:

con la collaborazione della risorsa Origin

- proposta della OpenURL
- ogni risorsa deve rendersi

OpenURL compliant cioè in grado di inviare i propri metadati



La OpenURL

OpenUrl

- E' un protocollo aperto, non proprietario
- è in elaborazione presso la NISO (accettata la proposta come v. 0.1 nel gennaio 2002, il draft della nuova versione uscito nel marzo 2003)
- si può considerare un **gancio** che la risorsa inserisce
 - è una URL che trasporta identificatori e/o metadati
 - li passa come parametri ad un “risolutore di OpenURL”, cioè un componente di servizio
 - tali parametri sono strutturati secondo una ben determinata sintassi

Esempi di OpenURL

- **Citazione** (come si trova in una risorsa informativa):

*Moll JR, Olive & M, Vinson C. Attractive interhelical electrostatic interactions in the proline- and acidic-rich region (PAR) leucine zipper subfamily preclude heterodimerization with other basic leucine zipper subfamilies. J Biol Chem. 2000 Nov 3 ; 275(44):34826-32.
doi:10.1074/jbc.M004545200*

- **Esempi di possibili OpenURL (codificate come richieste HTTP GET):**

<http://sfx1.exlibris-usa.com/demo?sid=ebsco:medline&aulast=Moll&auinit=JR&date=2000-11-03&stitle=J%20Biol%20Chem&volume=275&issue=44&spage=34826>

<http://sfxserv.rug.ac.be:8888/rug?id=doi:10.1074/jbc.M004545200>

La sintassi OpenURL

- È una richiesta HTTP GET (o POST)
- Base URL + Query
 - la base URL = indirizzo web del risolutore
 - Query = parametri per il risolutore
 - possono essere identificatori o metadati
 - espressi come NOME=VALORE
 - descrivono sempre la link-source
 - possono descrivere la risorsa origine

La sintassi della OpenURL v.0.1

QUERY ::= DESCRIPTION ('&&' DESCRIPTION)

DESCRIPTION descrive l'origine dell'oggetto di metadati trasportato + l'oggetto di metadati stesso.

Se ci sono più oggetti, le loro DESCRIPTION devono essere delimitate da 2 & (e commerciale).

**DESCRIPTION ::= (ORIGIN-DESCRIPTION '&')? OBJECT-DESCRIPTION | OBJECT-DESCRIPTION
('&' ORIGIN-DESCRIPTION)?**

(la QUERY può essere costituita dalla descrizione della *link-source* e, facoltativamente, della sua risorsa di origine, nell'ordine che si vuole)

ORIGIN-DESCRIPTION ::= sid '=' VendorID ':' databaseID

(altamente consigliato, obbligatorio se si usa la LOCAL-IDENTIFIER-ZONE)

OBJECT-DESCRIPTION ::= ZONE ('&' ZONE) *

(ci sono 3 tipi di ZONE possibili per descrivere l'oggetto, tutte facoltative, ma almeno una deve esserci)

**GLOBAL-IDENTIFIER-ZONE ::= 'id' '=' GLOBAL-NAMESPACE ':' GLOBAL-IDENTIFIER ('&' 'id' '='
GLOBAL-NAMESPACE ':' GLOBAL-IDENTIFIER)***

GLOBAL-NAMESPACE ::= ('doi' | 'pmid' | 'bibcode' | 'oai')

(può essercene più di uno: quelli ammessi per ora sono: **doi**, **pmid**, **bibcod**, **oai**)

OBJECT-METADATA-ZONE ::= META-TAG '=' META-VALUE (& META-TAG '=' META-VALUE) *

**(i META-TAG ammessi sono 'genre' | 'aulast' | 'aufirst' | 'auinit' | 'auinit1' | 'auinitm' |
'coden' | 'issn' | 'eissn' | 'isbn' | 'title' | 'stitle' | 'atitle' | 'volume' | 'part' | 'issue' | 'spage' |
'epage' | 'pages' | 'artnum' | 'sici' | 'bici' | 'ssn' | 'quarter' | 'date')**

LOCAL-IDENTIFIER-ZONE ::= 'pid' '=' VCHAR+

(se c'è, deve esserci anche una ORIGIN-DESCRIPTION)

[http://sfx1.exlibris-usa.com/demo?
sid=ebsco:medline&aulast=Moll&auinit=JR&date=2000-11-
03&stitle=J%20Biol%20Chem&volume=275&issue=44&spage=3
4826](http://sfx1.exlibris-usa.com/demo?sid=ebsco:medline&aulast=Moll&auinit=JR&date=2000-11-03&stitle=J%20Biol%20Chem&volume=275&issue=44&spage=34826)

- questa OpenURL permette dunque di definire tre entità:
 - il risolutore
 - il documento/oggetto della citazione bibliografica
 - la risorsa/origine da cui la citazione proviene
- L'oggetto bibliografico può essere descritto da vari *descrittori*
- ogni *descrittore* è espresso da coppie *nome=valore*.

Per capire meglio come è costruita
una OpenURL

si può giocare con il
“Generatore di OpenURL

<http://www.sfxit.com/OpenURL/generator.html>

<http://demo.exlibrisgroup.com:8888/OpenURL/article.html>

<http://linkserv.lib.utk.edu:9003/sfx>

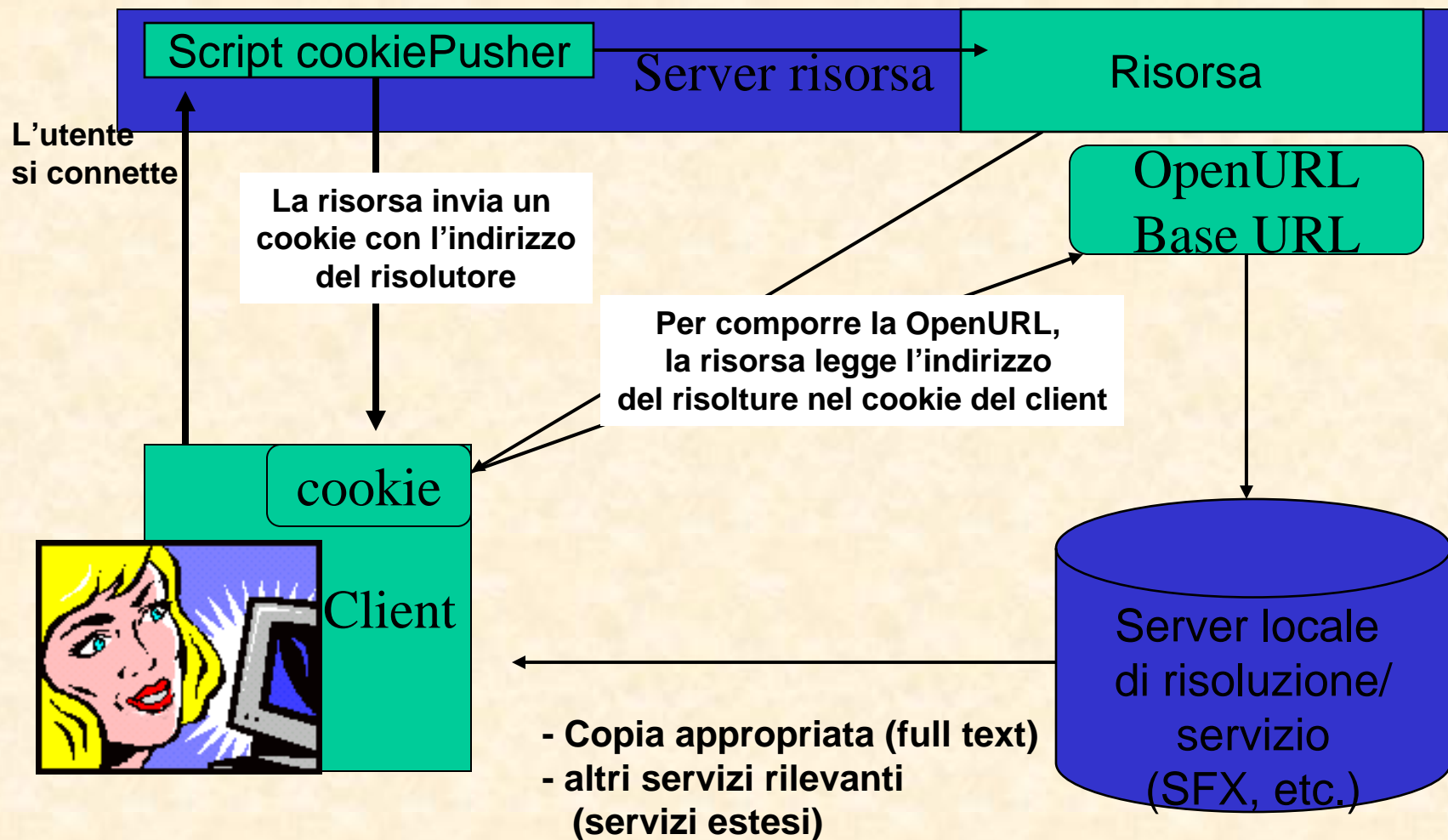
Per rendersi *OpenURL compliant*

- una risorsa *Origin* deve porsi in grado di emettere una *OpenURL*
 - opportunamente formattata
 - opportunamente indirizzata

CookiePusher: strumento di SFX awareness

- dispositivo che permette alla risorsa *OpenURL compliant* di conoscere l'indirizzo del risolutore dell'utente, da inserire nella BASE URL
- è uno script che la Risorsa mette sul suo server come primo accesso
- riceve 2 parametri
 - la url del componente di servizio (e la scrive in un cookie sul client)
 - la url della risorsa (a cui reindirizza la chiamata dell'utente)

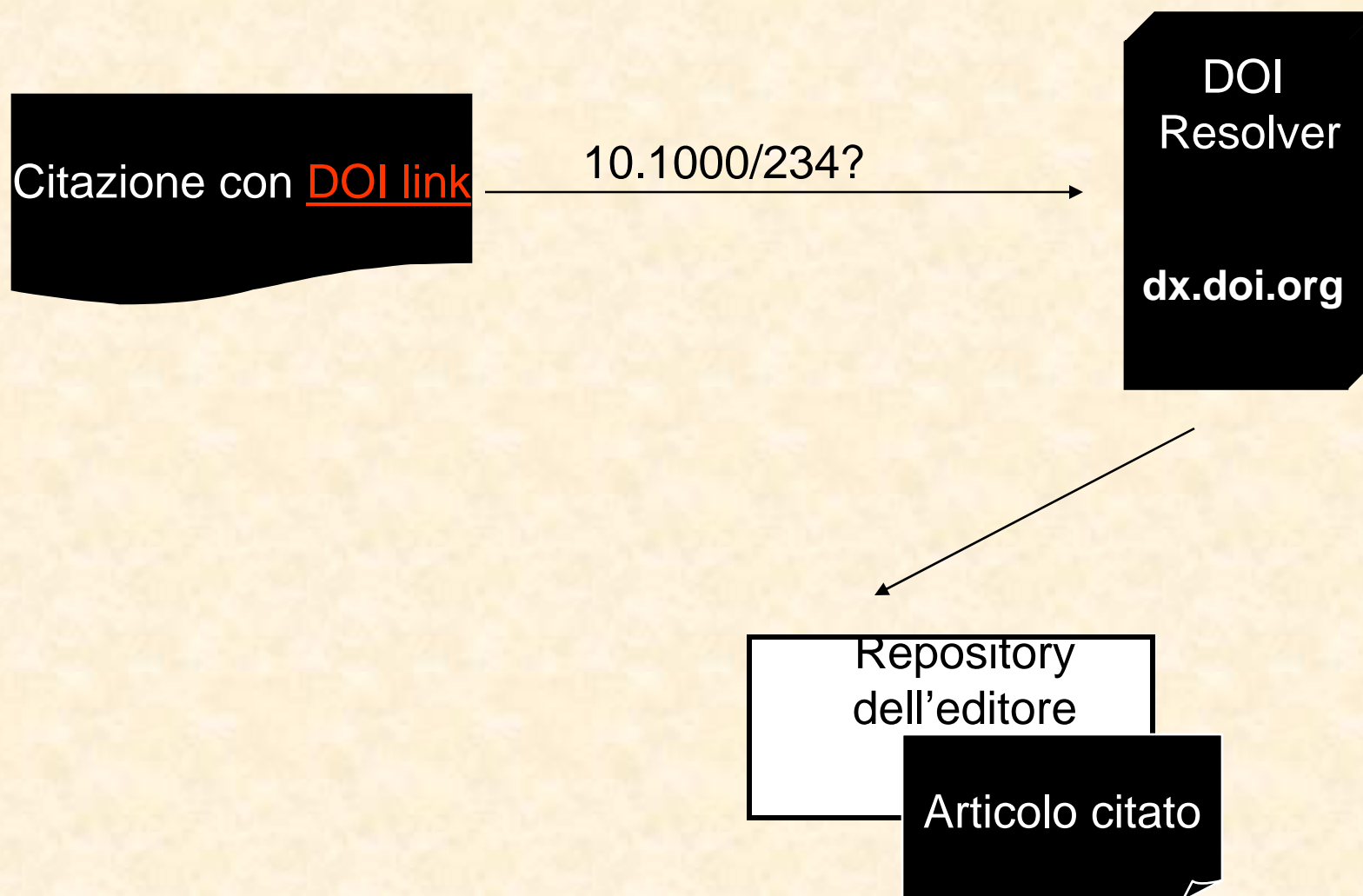
Una Risorsa resa “OpenURL compliant” dal suo Provider



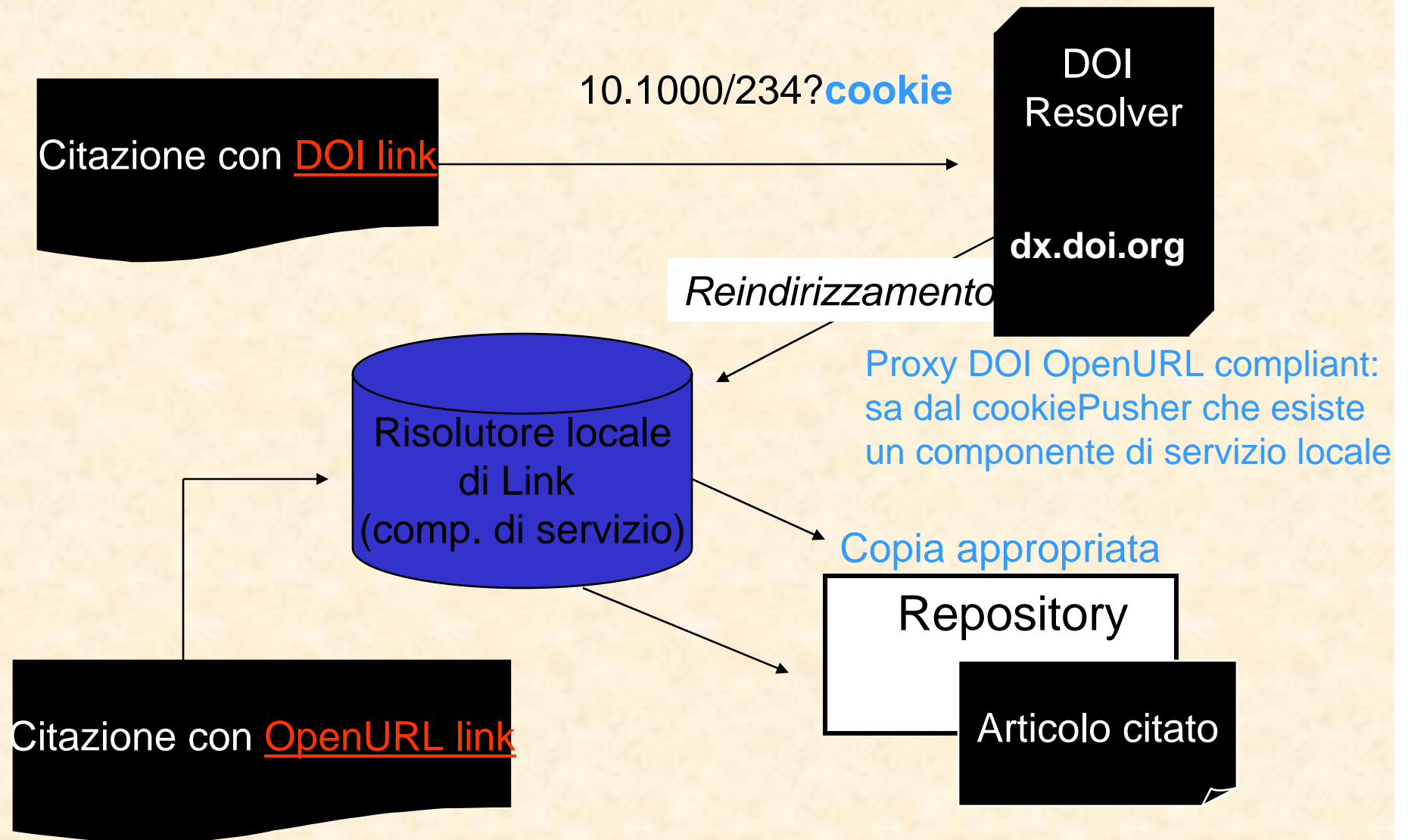
Collaborazione SFX / DOI

- DOI
 - sistema nato dall'iniziativa collaborativa di editori
 - per collegare citazioni e full-text di articoli
 - funzionante grazie a un server centrale, dove viene registrato, per ogni nuovo articolo:
 - i metadati
 - il codice DOI
 - la url del full-text (di default)
- non *context sensitive* inizialmente, si è reso *OpenURL compliant*

Come lavora il DOI non “localizzato”



Come lavora l'attuale "localizzazione" del DOI



La OpenURL nuova versione

- Draft della nuova versione uscito nel marzo 2003
http://library.caltech.edu/openurl/Working_Documents.htm
- per descrivere entità varie, non solo bibliografiche, su Web
 - anticipato nell'articolo sul modello Bison-Futé
 - problematiche tangenziali al web sematico

La nuova OpenURL

- non è posta in relazione con alcun componente di servizio
- non è vincolata al formato GET della URL, ma ipotizza anche l'uso dell'XML e di Schemi XML
- non è vincolata a *un* formato di metadati ma ammette l'uso di ogni formato di metadati

Il nuovo standard parla di:

- ***ContextObject***: un costrutto informativo che describe:
 - una *risorsa citata*
 - da 0 a molti *elementi contestuali*:
- ***Entità*** descritte all'interno del ContextObject:

• referent	(= oggetto descritto , risorsa citata)	(1, sempre)
• resolver	(= BASE URL)	(0 o 1)
• referrer	(= CHI invia la OpenURL , <i>Origin</i>)	(0 o 1)
• referring entity	(-> entità che fa la citazione, <i>Source</i>)	(0 o 1)
• requester	(-> CHI chiede i servizi (utente))	(da 0 a molti)
• service-type	(-> il tipo di servizio richiesto)	(da 0 a molti)
- ***Descrittore***: costrutto unitario per la descrizione di una entità

I *descrittori* possono essere di diverso tipo

- identificatore:
 - citazione del dominio + identificatore unico nel dominio (DOI, e.mail, URI)
- descrizione in metadati => *informaz. esplicita o “per valore”*:
 - citazione di uno schema + descrizione secondo quello schema
- puntatore ad una descrizione in metadati
=> *informazione implicita o “per citazione”*
 - citazione di uno schema + puntatore ad una descrizione dell'entità secondo quello schema
- zona privata

Formato di descrittore

- I *descrittori* possono essere codificati in modi diversi
- ogni tecnica di codifica costituisce un formato di descrittore
- un formato di descrittore si definisce in quanto in grado di citare in modo univoco gli schemi di metadati e di identificatori, insomma i *namespaces*

La nuova OpenURL è un framework generale, quindi estensibile, e prevede:

- la necessità di “registrare” i diversi formati e i metodi
- la necessità per ogni implementazione concreta di scegliere tra le molte possibilità
 - di fissare dei “profili di comunità” cioè dei sottoinsiemi dello standard adatti a particolari contesti

Il registro

<http://www.openurl.info/registry/>

1. [Namespaces](#)

- [URI Namespaces](#) (infrastruttura di Internet)
- [ORI Namespaces](#) (registrati per la OpenURL)
- [XRI Namespaces](#) (locali, non interpretabili globalmente)

2. [Character Encodings](#)

3. [Physical Representations](#)

4. [Constraint Languages](#)

5. [ContextObject Formats](#)

6. [Metadata Formats](#)

7. [Transports](#)

8. [Community Profiles](#)

Gli identificatori degli elementi nel Registro/1

1. Namespaces (URI e ORI)

- uri:urn:ISBN:<numero> uri:http://<url>
- ori:pmid:<codice di medline> ori:doi:<doi cod.>

2. Codifiche per i caratteri (UTF8, ISO-8859-1 etc.)

- ori:enc:UTF-8

3. Rappresentazioni fisiche del ContextObject (KEV, XML)

- ori:fmt:kev (formato Key/encoded value)
- ori:fmt:xml (formato XML)

4. Linguaggi di restrizione (per esprimere il precedente)

- ori:fmt:kev:mtx
- ori:fmt:xml:xsd

Gli identificatori degli elementi nel Registro/2

5. Formati del contextObject (costruiti su 3 e 4)

- ori:fmt:kev:mtx:**ctx**
- ori:fmt:xml:xsd:**ctx**

6. Formati di metadati (registraz.opzionale; costruiti su 3 e 4)

- uri:http://<url>
- ori:fmt:kev:mtx:book
- ori:fmt:xml:xsd:marc21

7. Metodi di trasporto: protocollo di rete e metodo d'impiego

- Ori:**tsf**:http:openurl01 (non all'interno del ContextObject)

8. Profili applicativi per comunità

OpenURL e SFX nascono insieme ma la OpenURL...

- può essere utilizzata in molti contesti, di servizio ma non solo
- OpenURL è:
 - uno standard per trasportare i metadati su web
 - un formato di metadati elaborabili dalla macchina

OpenURL e SFX nascono insieme ma il server SFX...

- è l'agente che implementa il linking *context sensitive*
- è un servizio di linking “intelligente”
- dove “intelligente” è fortemente legato alla ampia repertoriazione della Knowledge base SFX
- la quale è essenzialmente il valore aggiunto dai sistemi commerciali rispetto all'OpenResolver
 - contenuto della base
 - strumenti di gestione di essa

...ma il server SFX...

- l'architettura SFX proposta da Van De Sompel è una delle possibili
- rimane fortemente ancorata alle soluzioni *ad hoc* del prototipo, in gran parte precedenti alla OpenURL
 - p.e. gli script *sourceParser*, che pongono il Componente di Servizio in grado di “ri-accedere” alla *Origin* per arricchire i metadati ricevuti nella OpenURL, sono definiti per risorsa: più generalizzabile e in linea con lo standard OpenURL (per cui la definizione della *Origin* è opzionale) sarebbe un accesso a descrizioni centrali tramite identificatori standard (ora ciò può avvenire col DOI)

Che cosa si può fare con la sola OpenURL

- La OpenURL da sola non FA niente: è solo un contenitore di metadati
- È uno strumento su cui si può costruire un servizio (programmare un resolver/server etc.)

Che cosa fa l'OpenResolver free sw

- E' un programma .cgi che utilizza PERL e Javascript
- Va installato in un server web
- immessa una OpenURL in una *form*, attiva alcuni target generali (quali Amazon)

OpenResolver: raggiungere altri target

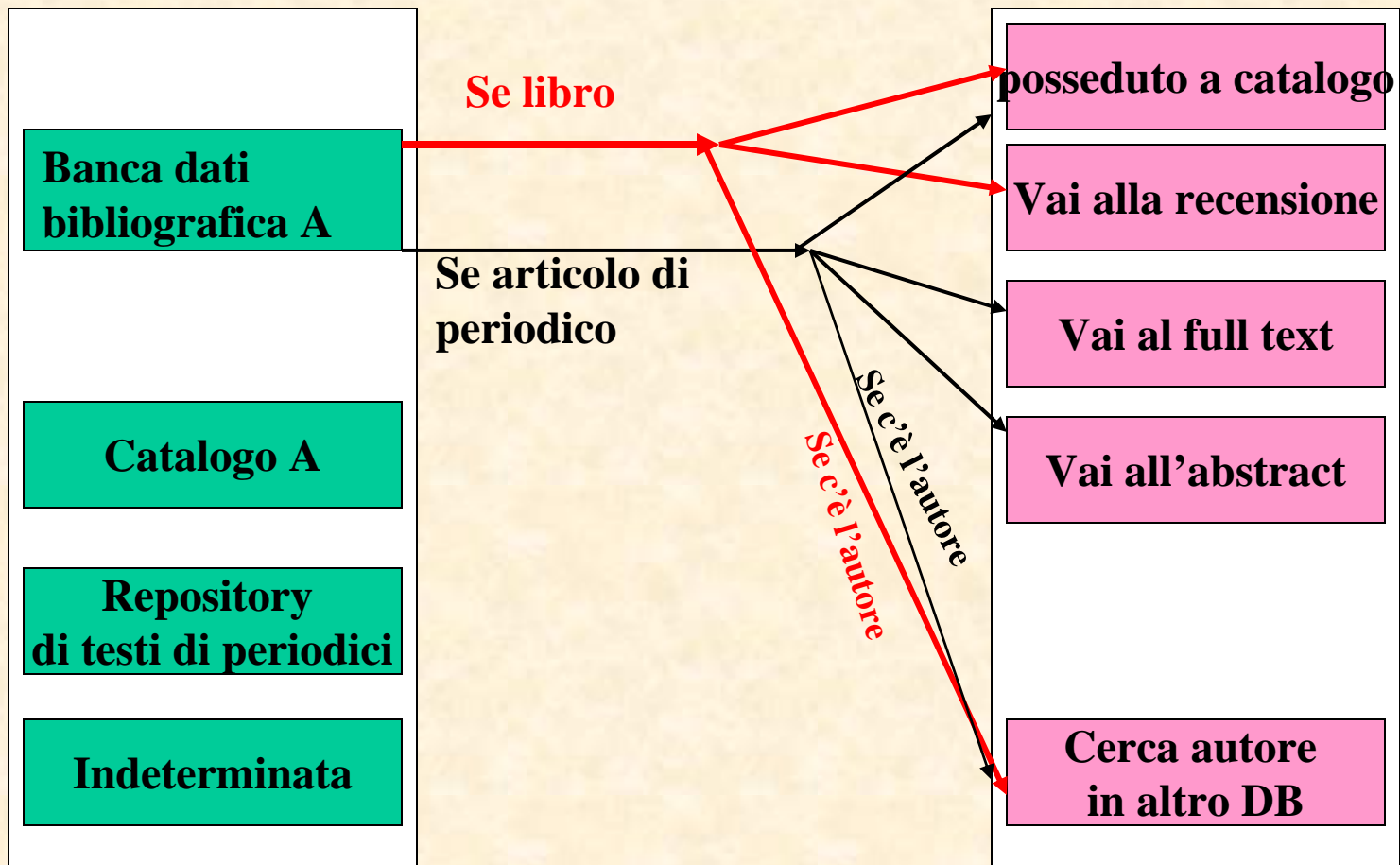
- È necessario implementare uno script per ogni singolo target, ad hoc (targetParser)
- e le soglie?
 - o si implementa una base repertoriale amplissima, che va strutturata e poi mantenuta (la Knowledge base!)
 - o si accettano un maggior numero di link a vuoto o irrilevanti

Dettagli:

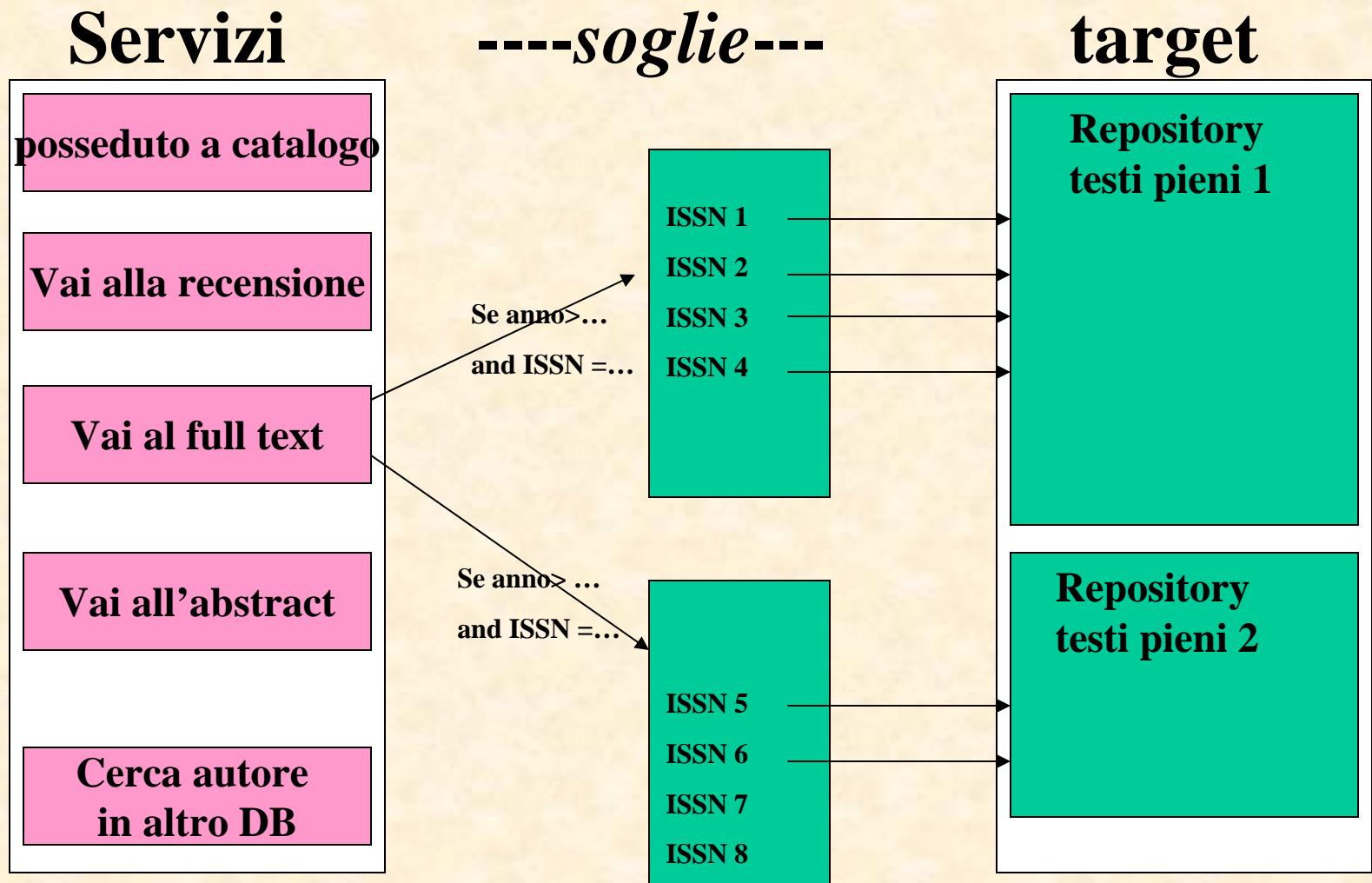
Origin/Sources

soglie

Servizi



Dettagli



Ricevere una OpenURL

- Implementato di solito in sistemi semichiusi:
 - riceve OpenURL da sistemi interni, che io costruisco
 - lo interfaccio col mio catalogo (reso *compliant*)
 - col mio server di e.print (reso *compliant*)
- problematica la ricezione di OpenURL esterne

Ricevere una OpenURL esterna

- che tipo di OpenURL vengono inviate dai vari servizi?

- lo standard prevede infatti anche OpenURL del tipo:

- <http://sfx1.exlibris-usa.com/demo?sid=ebsco:medline&pid=12345>

- _____ oppure

- [http://sfxserver.uni.edu/sfxmenu?sid=EBSCO:MFA&pid<author>Smit
h, Paul ; Klein, Calvin</author>&<yr>98</yr>](http://sfxserver.uni.edu/sfxmenu?sid=EBSCO:MFA&pid<author>Smit
h, Paul ; Klein, Calvin</author>&<yr>98</yr>)

- => necessario un accordo presentivo e un **sourceParser** ad hoc

- l'openURL può essere incompleta

- manca comunque di alcuni dati (autori successivi al primo, publisher)

- => servizio ridotto se non con **sourceParser** ad hoc

Concludendo

- Tutto dipende dall'ancora limitato processo di standardizzazione
- per ottenere la massima generalizzazione del servizio, è ancora necessario molto lavoro ad hoc
- quando si parte dal niente, può andare bene anche un livello di generalizzazione limitato
 - col vantaggio della massima personalizzazione

The end

**grazie
per l'attenzione
!**

Limiti del OpenResolver di UKOLN

- non è per ora in grado di reindirizzare i DOI
- non implementa ad un livello complesso una knowledge Base per filtrare i link irrilevanti
- non è chiaro a che livello sappia trarre i metadati dalla link-source, se gli viene passato solo l'identificatore privato: probabilmente lavora solo con namespace standard

Generalizzare l'applicazione ha comportato che:

La prima tabella COLLI è stata divisa in Servizi e Target

Il “componente di servizio” è stato diviso in “componente di reindirizzamento” e “componente di servizio”

Nella base SFX si è distinta: rilevanza globale vs rilevanza locale

L'introduz. della OpenURL, che rende facoltativa l'indicazione della *Origin*, ha comportato aggiustamenti nel COLLI quali l'introduzione relazione “da qualsiasi fonte” a “tutti i target”

Ancora un po' di terminologia

componente di servizio

il modulo software che fornisce servizi estesi sovrapposti

risolutore

nome più generico per il precedente (meno legato all'esperienza SFX)
per indicare un modulo software capace di ricevere una OpenURL e risolverla in servizi (in link)

sourceParser

uno script (un pezzo di sw) che rientra nell'architettura del componente di servizio, capace di analizzare una particolare link-source

targetParser

uno script (un pezzo di sw) che rientra nell'architettura del componente di servizio, capace di implementare il collegamento all'interno di un particolare target (una part. sintassi link-to)

Base SFX (Knowledge Base)

elemento centrale dell'architettura SFX: tabelle relazionali che registrano le conoscenze generali e contestuali necessarie al funzionamento context sensitive del componente di servizio